ISSN: 2393-8528

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 3 Issue 2; March-April-2016; Page No. 09-13

A Comparison of Database Query Languages: SQL, SPARQL, CQL, DMX

Kanchan Arora

M.Tech, IIIT, Delhi

kanchan.ar.aset@gmail.com

ARTICLE INFO

Received: 20 April. 2016
Accepted 30 April 2016

Corresponding Author:

Kanchan Arora

Email:
kanchan.ar.aset@gmail.com

Keywords:
SPARQL,SQL,CQL,DMX

ABSTRACT

In this paper, a comprehensive study of four database query languages has been provided. As we know, there are various types of database servers which store data in a format that is suited for particular needs for example, web servers are storing data in the form of web pages and online shopping database servers are storing data in the form of relational tables and if we go further to the linked data, facts are stored in the form of graphs. There are different types of database systems suiting different needs therefore multiple query languages designed to query appropriate databases are required to have some common properties and some special characteristics to fulfill purpose of their creation. For example queries for relational databases must have an appropriate mechanism to get data from multiple tables and languages which retrieve web pages must be able to provide ranking. This paper is an attempt to look closely and compare the essential common and special characteristics of widely query languages: SQL, SPARQL, CQL and DMX.

1. INTRODUCTION

In the early stages of databases, it was evident that file systems are not the best solution to store and process the huge amount of related data. In 1970 when E. Codd [1] introduced the concept of relational databases, it became a revolution in the field of computer science but the data in the database is useless if it cannot be queried. Since computers do not understand natural language, there is a need of query language through which users can fire queries on the database and can get the desired data. A query language is nothing but a set of commands that can be used to retrieve information from the database. SQL (Structured Query Language) was the language which was proposed in 1974 to query relational databases. SQL was basically the data retrieval language. But with the advancement in internet technologies and rapidly growing data on the web, there was a need of information query language which is not just capable of retrieving data on the web but can also rank the results while maintaining the simplicity as most users of the web generally prefer querying using natural language terms. CMX is one of the information query language which was developed to query web indexes. With the more advancement in web technologies, where Tim Berners Lee [4] introduced his vision of web in which machines can also interpret the meaning of data just like humans, RDF (Resource Description Framework)[3] databases were introduced. To query such meaningful data, in 2008 the query language named SPARQL [2] was developed. Many areas like weather forecasting, marketing, recommender systems require useful patterns to be mined from the raw data in order to have future predictions. In such cases, query language is required to train, browse and make predictions from data mining models. DMX (Data Mining Extension)[5] is one of the languages which can be used to create, maintain and query such models. In this paper the four query languages introduced above are discussed briefly and comparison based on certain parameters has been made.

The organization of this paper is as follows:
- In section II, III, IV and V general features of SQL, SPARQL, CQL and DMX respectively are discussed.
- In section VI, comparison facets along which the four query languages are compared are described briefly and a comparison table is provided to the users to decide which language to be used based on their requirement setting.
-In section VII, the paper is concluded by discussing the remarkable features and importance of each language.

II. SQL

SQL stands for **Structured Query Language**. It is a query language designed for retrieval and management of data in relational database. SQL is Data Definition, Data Manipulation and Data Control Language. Data can be defined using CREATE, ALTER and DROP commands. SELECT INSERT, UPDATE and DELETE commands can be used to retrieve and modify the data in the database. Permissions to the access the database can be given and taken from users by using GRANT and REVOKE commands. SQL also supports transaction control by providing ROLLBACK and COMMIT commands [11]. SQL is case-insensitive but case in identifiers makes a difference for e.g. table name "Employee" is different from "employee".

SQL supports exact numeric, approximate numeric, date, time, character strings, Unicode character strings and binary data types. Arithmetic operators(+,-,*,/,%), comparison operators(=, !=, <>, >, <, >=, <=,!<,!>) and logical operators (ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE,NOT,OR,ISNULL,UNIQUE) [11] can be used in SQL queries. Boolean expressions can also be used in SQL queries using WHERE clause to fetch the desired data. Numeric expressions can be used to perform mathematical operations and results can be displayed using AS clause. SQL also provides aggregate functions like avg(), count(), sum() to perform data calculations. SQL also supports wildcard operators like % and _ (underscore). TOP and LIMIT clause are also available to fetch the limited number of records. The data retrieved can be sorted in ascending and descending order using ORDER BY clause. Identical data can be arranged into groups by using GROUP BY clause. Unique records can be fetched using DISTINCT keyword in the query. SQL can also be used to add and drop constraints.

Data from multiple tables can also be fetched using SQL. SQL Joins are used to combine records from multiple tables. INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, SELF JOIN, CARTESIAN JOIN are the joins available in SQL. The results from two or more SQL queries can also be combined using UNION and UNION ALL, INTERSECT and EXCEPT clause as desired. SQL also provides mechanism to represent a missing value i.e. a field whose value is unknown at the time of entering the data, that field can be filled with NULL value. It is neither zero nor blank space. Although it creates problems while retrieving data with conditional queries but it helps to enter a record whose one field value may be unknown.

A table or column can be renamed by the use of aliases. Indexes can be created on tables using CREATE INDEX commands. Indexes speed up the SQL queries. Once a table has been created and filled in with data, its structure can be altered using ALTER TABLE command. TRUNCATE and DELETE command can be used to delete data from the tables. SQL also supports the concept of view. View is an SQL statement stored in database as a name but it behaves as a kind of virtual tables. Temporary tables can also be created using SQL in RDBMSs which they are supported. These tables can be used till session is alive. Sub-queries can be used in SQL SELECT, INSERT, UPDATE and DELETE statements. Relational database management systems like MYSQL, Oracle, Postgres and SQL Server use SQL to query and maintain data in database.

III. SPARQL

SPARQL stands for **S**PARQL **P**rotocol and **R**DF **Q**uery **L**anguage. It is a query language which is used to query RDF data. RDF data is in the form of triples: <subject><predicate><object>. If one of the three is known, the other two can be queried. RDF captures entity attributes and relationships between entities through triple statements. Most of the keywords in SPARQL have been taken from SQL as all the operations which can be performed using SQL can also be performed using SPARQL but there are certain notable differences between the two. One of the differences is that there is no notion of NULL value in SPARQL. A tuple with missing data is simply not added in the database. The other difference is LEFT OUTER JOIN in SQL is performed using OPTIONAL keyword [13]. In SPARQL graph existence of a graph pattern can also be searched for using ASK keyword. Pattern Matching can be performed by using regex [12]. In relational database each column has homogenous data and therefore care of data types must be taken while entering the data but in RDF database each subject or object have heterogeneous data therefore there is no need of data type alignment. Data from multiple databases can be integrated and queried using SPARQL. In cases where data is scattered in multiple tables and to get the required data multiple table join has to be performed, SQL queries reflect the structure of tables, information about keys and how the data is stored in multiple tables. But, in such cases in SPARQL queries focus is more on the semantics [14].

IV. CQL

CQL stands for **Contextual Query Language**. It is a formal language which is simple yet expressive for querying information retrieval systems. A CQL query may have a single search clause or multiple search clauses combined by Boolean operators. A search clause may be single term that is to be searched in documents or may consist of index, relation and search term. An index is defined as a part of context set. For e.g. the index 'title' belongs to context set 'dc'. Each occurrence of indexes, relations, relational modifiers, boolean modifiers and index modifiers belong to a context set. CQL uses context sets to define community specific semantics. A search term must be enclosed in double quotes if it contains <,>,(,)=,\,",' ' . A relation specifies the relationship between index and the search term. By default it is assumed to be =. The relational operators supported by CQL are =, <, >, <=, >= , <> , ANY and RELEVANT. The boolean operators that can be used in CQL are AND, OR, NOT, PROX [8]. The prox operator is used when the relative locations of the terms are also specified and desired in the result set. The proximity unit can be a word or a sentence. The result set generated by the search can be sorted by sortBy keyword. CQL is case-insensitive except for search terms and modifiers' values. The data types that can be used in CQL are word, strings, number, date, URI. The wild card characters (*, ?, ^) are also supported for matching of strings. Boolean operators also include modifiers such as ordered and unordered. ORDERED is used when the order of search terms in results should be same as it is in query and UNORDERED is used when order of terms is immaterial [8].

V. DMX

DMX stands for **Data Mining Extensions**. It is a language that can be used to create and work with data mining models. Using DMX, mining structures can be created for new data mining models. These structures can be used to train models, browse and manage them. The language can also be used to make predictions from the trained models. Since DMX can be used to create and train data models, it is both data manipulation and data definition language.

Data Definition statements include CREATES MINING STRUCTURE, CREATE MINING MODEL, ALTER MINING STRUCTURE, DROP MINING STRUCTURE, IMPORT and EXPORT model and structures. Structure of an existing model can also be copied into new model by using SELECT INTO statement [6].

Data Manipulation statements are used when it is required to browse the existing models and to make predictions from them. An existing model can be trained by using INSERT INTO statement. Existing models can be browsed by using SELECT statement. PREDICTION JOIN statement can be used to make predictions. Training data in models can be deleted using DELETE statement.

A DMX expression is a combination of identifier, values and operators. A value can be a string, numeric or data. DMX operators are of four types: Arithmetic operators (+,-,*,/), Logical operators (AND,OR,NOT), Comparison operators (>,<,=,<>,>=,<=) and unary operators(+,-) [6]. In DMX, functions can be used to return predicted values of a column as well as other information like statistics about predictions [6]. There are two types of DMX functions scalar and non-scalar. Scalar return single value and non-scalar return a table. DMX supports NULL values and hence have a three-valued logic. Data types allowed in DMX queries are Numeric, String, Date and Boolean. There are content types which defines the behaviour of data in a column. There are Discrete, Continuous, Discretized, Key, Key Sequence, Key Time, Table, Cyclical, Ordered, Classified content types available [9]. The SESSION keyword can be used to create temporary structures that will last only for current session [6].

-

VI. COMPARISON

In this section, comparison of the four query languages described in previous sections is provided along certain parameters. A brief description of the parameters is provided first and then a comparison is provided in Table1 for the readers to have a close look at whether the corresponding features are present in a query language or not.

The following facets have been chosen to compare SQL, SPARQL, CQL and DMX:

- **Purpose-of-built**: It describes that a given language is capable of querying what type of database (relational or RDF or web indexes or data mining models).

- **Declarative or procedural:** If the query language describes how the required data is to be fetched from database it is procedural and if it only specifies the required data then it is declarative. Query languages should be declarative.

- **Joins:** It describes whether the query language is capable of retrieving data from multiple data units. A query language is considered to be robust and expressive if it can provide such mechanism.

- **Base:** It describes that n the process of execution of a query, to what type of expression, it is converted into to get optimized execution. SQL queries are converted into relational algebra expressions before going into the query optimisation phase.

- **Logical Operators:** It describes what are the logical operators supported by the language and hence deciding how much complex logical expressions can be articulated using the given language.

- **Comparison Operators:** It describes what operators are available for the expressions which require comparison of identifiers and values.

- **Arithmetic Operators:** It describes what type of mathematical expressions can be articulated by the language thereby determining the expressive power of a language.

Table1: Comparison Table

Comparison Facet	Comparison Table			
	SQL	***SPARQL***	***CQL***	***DMX***
Created In	1974	2008[7]	2003	2012
Developed By	ISO/IEC[10]	W3C	Z39.50 Maintenance Agency[15]	Microsoft
Type	Database query language	Graph Matching Query Language	Information Retrieval Query Language	Data Mining Model Language
Purpose of built	To query Relational DataBases	To query RDF Databases	To query Web Indexes, Catalogues	To query Mining Models and Mining structures
Declarative/Procedural	Declarative and Procedural[10]	Declarative	Declarative	Declarative
Joins	INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, SELF JOIN, CARTESIAN JOIN[11]	Only Left outer join using OPTIONAL keyword[13]	NIL	Prediction join
Based On	Relational Algebra and Tuple Relational Calculus[10]	Relational Algebra	Boolean Algebra	Relational Algebra
Logical Operators Supported	ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE,NOT, OR,ISNULL, UNIQUE[11]	&&,||	AND, OR, NOT, PROX	AND, OR, NOT
Arithmetic Operators Supported	+,-,*,/,%	+,-,*,/	NIL	+,-,*,/
Comparison Operators	=, !=, <>, >, <, >=, <=,!<,!>	=, !=, , >, <, >=, <=	= , >, <, >=, <=,<>,==	>,<,=,<>,>=,<=
Expressions supported	Boolean,Numeric,Date	Boolean, Numeric, Date	Boolean	String, Numeric, Date
Modifiers Supported	LIMIT, DISTINCT,ORDER BY	PROJECTION, DISTINCT,ORDER,LIMIT,OFFSETS,REDUCED	sortBy, ordered, unordered	DISTINCT, TOP, ORDERBY
DataTypes Supported	Exact numeric, approximate numeric, datetime, character strings, Unicode character strings and binary data types	Numeric, string, boolean, datetime	Word ,Numeric, strings, date, URI	Numeric, String ,Date, Boolean, Content Types: Discrete, Continuous, Discretized, Key, Key Sequence, Key Time, Table, Cyclical, Ordered, Classified[9]
Conjunction Operator	AND	No keyword for conjunction	AND	AND
Logic used	Three valued Logic(True,False,Unknown)[10]	Two-valued logic	Two-valued logic	Three-valued logic
NULL values supported	YES	NO	NO	YES
Whether Data Manipulation Supported	YES	YES	NO	YES
Whether Data Control Supported	YES(thorugh Grant and Revoke)	NO	NO	NO
Whether Data Definition Supported	YES	YES	NO	YES
Case Senstivity	Case-Insensitive	Case-Insensitive	Case-insensitive	Case-Insensitive
Temporary Data	Temporary tables can be created using TEMPORARY keyword with create command	No such support as SPARQL is a stateless protocol	No such support	Temporary mining structures can be created using SESSION keyword
Extensions	PL-SQL(Cursors, Triggers, Functions)	GEOSPARQL, SPARUL[7]	NIL	NIL
Remarkable Features	JOINS, Transaction Management	FILTERS, Ability to query from multiple databases	Proximity Modifiers	Functions and Predictions

- **Expressions:** It describes what types of expressions are supported by the language. For example Boolean, numeric and date. It clearly defines how expressive a given query language is.
- **Modifiers:** It describes whether the results obtained can be formatted or viewed in a desired fashion for e.g. may be in particular sorted order or only few results from the whole result set is desired.
- **DataTypes:** This describes what type of data can be handled by the queries. For e.g. integers, strings, Boolean, datetime etc.
- **Logic:** It describes what values a Boolean expression in a query can return. If it can only return true and false, it is two-valued logic and it can return true, false and unknown value then it is three-valued logic.
- **NULL values:** It describes whether a language has any mechanism to handle values which are unknown at time of both data definition and data manipulation.
- **Data Control:** It states whether the language provide authorization features or not.
- **Data Manipulation:** It states whether the language is able to update the data in database or not.
- **Data Definition:** It states whether the language can define the schema of the database.
- **Case Sensitivity:** It describes whether the keywords can be typed either in lowercase or upper case making it simple and user-friendly.
- **Temporary data:** It describes whether there is any provision to create a data unit which is only available for the current session.
- **Extensions:** This describes whether there are languages which are built based on the features of the given language.
- **Remarkable Features:** This describes the incomparable features of the language which makes the language acceptable for their particular use.

VII. CONCLUSIONS

In this paper a comprehensive comparison of four database query languages has been provided. A query language irrespective of type of database it is querying, it should be user-friendly, efficient, simple yet expressive. Case Insensitivity and its support for data manipulation and definition make it more user-friendly. The efficiency of the language depends upon the underlying expression it will be converted to and query optimization process. Expressive power of language can be determined by type of expressions, operators, data types it allows. Expressive power of SQL is phenomenal, as it is able to put across almost every data need of the user. Support for NULL values, sub-queries, multiple table joins and the way transactions can be managed are some of the unique features of SQL making it the most widely used relational database query language. SPARQL can be used both for relational and RDF graph database. SPARQL can also be used to query multiple RDF databases. CML being information retrieval language does not require data manipulation. DMX is associated with data mining have mechanisms to make predictions.

REFERENCES:

1. Edgar F. Codd. A relational model of data for large shared data banks. Communication of the ACM, 13(6):377–387, 1970.
2. B. Quilitz and U. Leser, Querying Distributed RDF Data Sources with SPARQL, In proceedings of the 5th European Semantic Web Conference on The Semantic Web:Research and Applications, ESWC’08, pages 524-538, 2008.
3. R. Cyganik, D. Wood and M. Lanthaler, RDF 1.1 Concepts and Abstract syntax, 2014.
4. T.Berners-Lee,J. Hendler and O. Lassila, The Semantic Web, Scientific American,284(5):29-37,5, 2001.
5. https://en.wikipedia.org/wiki/Data_Mining_Extensions
6. http://download.microsoft.com/download/0/F/B/0FBFAA46-2BFD-478F-8E56-7BF3C672DF9D/Data%20Mining%20Extensions%20-%20DMX%20-%20Reference.pdf
7. https://en.wikipedia.org/wiki/SPARQL
8. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/os/part5-cql/searchRetrieve-v1.0-os-part5-cql.html
9. https://msdn.microsoft.com/en-in/library/ms174572.aspx
10. https://en.wikipedia.org/wiki/SQL
11. http://www.tutorialspoint.com/sql/
12. https://www.w3.org/2001/sw/DataAccess/rq23/
13. http://www.cambridgesemantics.com/semantic-university/sparql-vs-sql-intro
14. http://www.topquadrant.com/2014/05/05/comparing-sparql-with-sql/
15. https://en.wikipedia.org/wiki/Contextual_Query_Language